Joint Concept Learning and Semantic Parsing from Natural Language Explanations

Shashank Srivastava **Igor Labutov** **Tom Mitchell**
Machine Learning Department
Carnegie Mellon University
Pittsburgh, PA 15217
ssrivastava@cmu.edu ilabutov@cs.cmu.edu tom.mitchell@cmu.edu

Abstract

Natural language constitutes a predominant medium for much of human learning and pedagogy. We consider the problem of concept learning from natural language explanations, and a small number of labeled examples of the concept. For example, in learning the concept of a phishing email, one might say 'this is a phishing email because it asks for your bank account number'. Solving this problem involves both learning to interpret open-ended natural language statements, as well as learning the concept itself. We present a joint model for (1) language interpretation (semantic parsing) and (2) concept learning (classification) that does not require labeling statements with logical forms. Instead, the model prefers discriminative interpretations of statements in context of observable features of the data as a weak signal for parsing. On a dataset of email-related concepts, this approach yields across-the-board improvements in classification performance, with a 30% relative improvement in F1 score over competitive classification methods in the low data regime.

1 Introduction

The ability to automatically learn concepts[1] from examples is a core cognitive ability, with applications across diverse domains. Examples of such concepts include the concept of a 'negative review' in product reviews, the concept of 'check' over the domain of game states in chess, the concept of 'fraud' in credit history analysis, etc. Concept learning is generally approached using classification methods that can automatically leverage regularities in large amounts of labeled training data. However, there are two shortcomings of this paradigm. First, labeling large amounts of data is unnatural compared to how a person might teach another person (e.g., a human secretary) in a similar situation. For example, for identifying emails about postdoc positions, a university professor might say 'These inquiries usually seek a postdoc opportunity and include a CV', rather than label scores of examples of such emails. Second, acquiring large quantities of labeled data may be infeasible because of a long tail of concepts that are highly domain or user specific. For our example of a busy professor, it might be relevant to teach concepts such as 'postdoc seeking emails', 'course related questions from students', etc. to an email assistant in order to better manage her/his inbox. However, these concepts might be irrelevant to a general user.

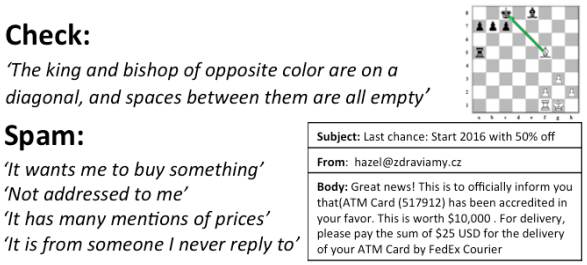

Figure 1: Examples of concepts explained using natural language statements.

On the other hand, humans can efficiently learn about new concepts and phenomena through language. In fact, verbal and written language form the basis for much of human learning and pedagogy, as reflected in text-books, lectures and student-teacher dialogues. Natural language explanations can be a potent mode of supervision, and can alleviate issues of data sparsity by directly encoding relevant knowledge about concepts. Figure 1 shows

[1] where a concept is any Boolean function on some domain of instances.

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1528–1537
Copenhagen, Denmark, September 7–11, 2017.

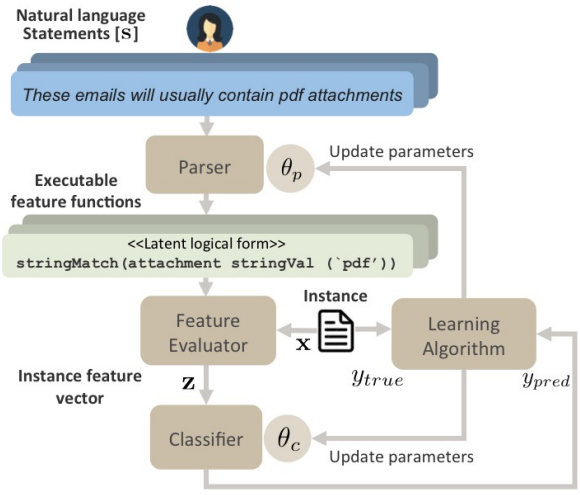

Figure 2: Schematic representation of approach

examples of concepts explained using natural language. In general, natural language can subsume several modes of supervision: instance labeling (e.g., 'This email is spam'), feature labeling (e.g., 'The word 'Viagra' indicates spam'), model expectations ('Spam emails *rarely* come from edu extensions'), etc. However, here we focus on the ability of natural language to express rich and compositional features for characterizing concepts.

In this paper, we address the task of learning concepts from natural language statements and a small number of labeled examples of the concept. Figure 2 summarizes the outline of our approach. We map statements to logical interpretations, which can be evaluated in context of new instances. In doing this, each statement s effectively acts as a binary feature function $\{z = f_s(x) \in \{0, 1\}\}$ that fires when the interpretation of a statement s is true for an instance x. The crux of our approach is that correct interpretations of natural language explanations are more likely to be useful in discriminating concepts, and this observation can be used to guide both semantic interpretation and concept learning[2].

In Section 3, we describe our probabilistic latent variable formulation that learns a semantic parser and a concept classifier from labeled examples of the concept. The latent variables correspond to evaluations of natural language statements for different instances, and training proceeds via a generalized EM procedure that iteratively (1) estimates evaluations of explanations (marginalizing over all interpretations), and (2) updates the classification and semantic parsing models. The inputs to the method consist of a small number of labeled examples and non-examples of a concept, natural language statements explaining the concept, and a domain specific lexicon. The method does not require labeling sentences with logical forms.

For our empirical evaluation, we focus on personal emails, a practical example of a domain where target concepts are often highly individualized and labeled data is scarce. The contributions of this work are:

- We introduce the problem of concept learning from natural language. We also collect a corpus of emails about common email concepts, along with statements from human users explaining these concepts.
- We provide a method for concept learning and language understanding that can be trained from a small number of labeled concept instances. Thus, we extend supervised semantic parsing by learning from a weaker form of supervision than has previously been explored.
- We demonstrate that for small labeled data, using natural language statements can achieve substantial gains in classification accuracy.

2 Related work

Concept learning from labeled examples has been a dominant focus of research in supervised learning (Caruana et al., 2008). Notable approaches such as Generalized Expectation (Mann and McCallum, 2010) and Posterior Regularization (Ganchev et al., 2010) have explored integration of manually provided 'side-information' (feature and label constraints) to guide machine learning models. Earlier work on Explanation-based learning (Mitchell et al., 1986; DeJong and Mooney, 1986) leverages structured knowledge to 'explain' why an example belongs to a concept. Recent work by Lake et al. (2015) explores visual concept learning from few examples, and presents encouraging results for one-shot learning by learning representations over Bayesian programs. However, none of these address the issue of learning from natural language.

Semantic interpretation of language has been explored in diverse domains. While semantic parsers have traditionally relied on labeled datasets of statements paired with labeled logical forms (Zettlemoyer and Collins, 2005), recent approaches have focused on training semantic parsers from

[2]e.g., a parser may associate multiple incorrect interpretations with the statement in Figure 2 (like `stringMatch(attachment stringVal ('usually'))`), which are unlikely to help in discriminating instances of the concept.

denotations of logical forms, rather than logical forms themselves (Krishnamurthy and Mitchell, 2012; Berant et al., 2013). Our work extends this paradigm by attempting to learn from still weaker signal, where denotations (evaluations) of logical forms too are not directly observed. Similar to our work, previous approaches have used different kinds of external-world signals to guide semantic interpretation (Liang et al., 2009; Branavan et al., 2009). Natural instructions have been studied in game playing frameworks (Branavan et al., 2012; Eisenstein et al., 2009). Our work is also closely related to work by Goldwasser and Roth (2014); Clarke et al. (2010), who also train semantic parsers in weakly supervised contexts, where language interpretation is integrated in real-world tasks. The general idea of learning through human interactions has previously been explored in settings such as behavioral programming (Harel et al., 2012), natural language programming (Biermann, 1983), learning by instruction (Azaria et al., 2016), etc. To the best of our knowledge, this work is the first to use semantic interpretation to guide concept learning.

3 Method

We consider concept learning problems in which the goal is to approximate an unknown classification function $f : X \rightarrow Y$ where $Y = \{0, 1\}$. The input to our learning algorithm consists of a set of labeled training examples $\mathcal{T} := \{(x_1, y_1), \ldots, (x_m, y_m)\}$, along with a set of natural language statements $\mathcal{S} := \{s_1 \ldots s_n\}$ about the concept. Our aim is to leverage statements in $\mathcal{S}$ to learn a better classifier for the concept. Our training data does not contain any other form of supervision (such as logical forms).

	s_1	s_2	s_j	s_m	Label
x_1	z_{11}	z_{12}	z_{1j}	z_{1m}	y_1
...					...
x_i	z_{i1}	z_{i2}	z_{ij}	z_{im}	y_i
...					...
x_n	z_{n1}	z_{n2}	z_{nj}	z_{nm}	y_n

Figure 3: Our data consist of instances x_i with binary labels y_i and statements $s_1 \ldots s_n$ about a concept. z_{ij} denotes whether the statement s_j applies to instance x_i, and is not observed in the data.

We assume that each statement s_j defines some Boolean property over the instances X; that is, statement s_j should be interpreted as defining a predicate $l_j : X \rightarrow \{0, 1\}$. We augment the representation of each instance, x_i, with a feature vector $\mathbf{z}_i$, that encodes the information contained in $\mathcal{S}$. The individual elements of this feature vector, $z_{ij} \in \{0, 1\}$, denote whether the statement s_j applies to instance x_i (see Figure 3). In the general case, the evaluation values $\mathbf{z}_i$'s are not directly observed. These are obtained by parsing each statement s_j into a logical expression $l_j : X \rightarrow \{0, 1\}$ which can be evaluated for an instance x_i to obtain $z_{ij} = [\![l_j]\!]_{x_i}$. Details of this evaluation are given in Section 3.4.

In this paper, we jointly learn a classifier and a semantic parser while treating z's as latent variables. For training, we maximize the conditional log likelihood of the observed data. Let us consider the log likelihood for a single data instance (ignoring the subscript i) for now. Since the evaluations $\mathbf{z}$ of natural statements for any context are latent, we marginalize over these. Using Jensen's inequality, any distribution q over the latent variables provides a lower-bound on the data log-likelihood:

$$\begin{aligned}
&\log p(y \mid x, \mathcal{S}) = \log \sum_{\mathbf{z}} p(y, \mathbf{z} \mid x, \mathcal{S}) \\
&\geq \sum_{\mathbf{z}} q(\mathbf{z}) \log \frac{p(y, \mathbf{z} \mid x, \mathcal{S})}{q(\mathbf{z})} \\
&= \sum_{\mathbf{z}} q(\mathbf{z}) \big(\underbrace{\log p_{\theta_c}(y \mid \mathbf{z}, x)}_{\text{classification}} + \underbrace{\log p_{\theta_p}(\mathbf{z} \mid x, \mathcal{S})}_{\text{parsing}} \big) \\
&\qquad + \mathcal{H}_q
\end{aligned} \tag{1}$$

Here, $\mathcal{H}_q$ is the entropy term for the distribution q.

3.1 Coupling parsing and classification:

In Equation 1, we observe that the data likelihood decouples into the log probability of observing the concept labels $p_{\theta_c}(y_i \mid \mathbf{z}, x)$ conditioned on the statement evaluations and the log probability of the latent statement evaluations $p_{\theta_p}(\mathbf{z} \mid x, \mathcal{S})$. In particular, the first term can be naturally parametrized by a discriminative classifier such as a loglinear model (with associated parameters θ_c). We provide more details in Section 3.3.

On the other hand, the probability of the latent statement evaluation values $\mathbf{z}$ can be parametrized using a probabilistic semantic parsing model (with associated parameters θ_p). The second term decouples over evaluations of individual statements ($\log p_{\theta_p}(\mathbf{z} \mid x, \mathcal{S}) = \sum_j \log p_{\theta_p}(z_j \mid x, s_j)$). In

turn, since we never observe the correct interpretation l for any statement, but only model its evaluation z_j, we marginalize over all interpretations whose evaluations in a context x matches z_j (similar to Liang et al. (2011)).

$$\log p_{\theta_p}(z_j \mid x, s_j) = \log \sum_{l:[\![l]\!]_x = z_j} p_{\theta_p}(l \mid s_j) \quad (2)$$

Following recent work in semantic parsing (Liang and Potts, 2015; Krishnamurthy and Mitchell, 2012), we use a log-linear model over logical forms:

$$p_{\theta_p}(l \mid s) \propto \exp({\theta_p}^T \phi(s, l)) \quad (3)$$

where $\phi(s, l) \in \mathbb{R}^d$ is a feature vector over statements s and logical interpretations l.

3.2 Learning:

In Equation 1, $q(\mathbf{z})$ denotes a distribution over evaluation values of statements; whereas θ_c and θ_p denote the model parameters for the classifier and semantic parser. The learning algorithm consists of an iterative generalized EM procedure, which can be interpreted as a block-coordinate ascent in the estimates of statement evaluations $q(\mathbf{z})$ and the model parameters θ_c and θ_p.

E-step: In the E-step, we update our estimates of evaluation variables ($\mathbf{z}$). We make a mean-field approximation by assuming that the joint distribution over evaluations decouples as $q(\mathbf{z}) = \prod q_j(z_j)$. Then maximizing the lower bound in Equation 1 in terms of q_j leads to the following update:

$$q_j(z_j) \propto \exp\left(\mathop{\mathbb{E}}_{j' \neq j} [\log p_{\theta_c}(\mathbf{z}|x)] + \log p_{\theta_p}(z_j|x, s_j) \right) \quad (4)$$

The first term in the update prefers values of an evaluation variable that are more discriminative on average (when values of other statements are marginalized out). The second term favours values of the evaluation variable that conforms with the most likely interpretations of the corresponding statement (s_j) by the semantic parser. Thus, in the E-step, we upweight evaluations of statements that are both discriminative, as well as supported by interpretations from the semantic parser.

M-step: In the M-step, we update the model parameters to maximize the lower bound in Equation 1. This corresponds to independently optimizing the log likelihoods for the classification model and the semantic parser, based on current estimates of $q_j(z_j)$'s of the statement evaluations. The entropy term H_q is constant from the perspective of model parameters, and is not relevant for the optimization. In particular, the semantic parser is updated to agree with evaluations of natural language statements that are discriminative. At the same time, the classification model is updated to fit evaluations that are supported by interpretations from the semantic parser.

We now describe the M-step updates for the log-linear semantic parser with parameters, θ_p. The updates for the classifier parameters, θ_c, depend on the form of the classification model, and are described in Section 3.3. For clarity, we focus on updates corresponding to a particular statement s_j from the training dataset. From Equations 1, 2 and 3, the objective for the semantic parser is given by:

$$\ell_j(\theta_p) = \sum_i \sum_{z \in \{0,1\}} q(z_{ij}) \log \frac{\sum_{l:[\![l]\!]_{x_i} = z} \exp({\theta_p}^T \phi(s_j, l))}{\sum_l \exp({\theta_p}^T \phi(s_j, l))} \quad (5)$$

Semantic parsers are usually optimized using gradient updates. Here, the gradient is:

$$\nabla \ell_j(\theta_p) = \sum_{i,z,l} q(z_{ij}) p_{\theta_p}(l \mid s) \frac{p_{\theta_p}(z_{ij} \neq z|x_i, s)}{p_{\theta_p}(z_{ij} = z|x_i, s)} \phi(s_j, l) \quad (6)$$

3.3 Classification models

The model and learning procedure described in Sections 3.1 and 3.2 is agnostic to the choice of the classification model (with parameters θ_c). For this work, we experimented with a logistic classifier (LR) and a Naive Bayes model (NB). We briefly describe these here:

Logistic Regression (LR): The form of the logistic function $\log p(y|\mathbf{z}) = -\log(1 + \exp(\text{-}\theta_c^T \mathbf{z}\ y))$ means that the likelihood does not decouple for individual components in $\mathbf{z}$. Hence, in the E-step, the expectation in Equation 4 cannot be computed analytically. Instead, we estimate this by drawing Bernoulli samples for individual z_j's using previous estimates of $q_j(z_j)$. In the M-step, we update classification parameters θ_c using stochastic gradient updates, while again sampling individual z_j's.

Naive Bayes (NB): The likelihood for this model is $p(y, \mathbf{z}) = \prod_j \theta_{cy}^{z_j} (1 - \theta_{cy})^{1 - z_j}$. In this case, the individual components of $\mathbf{z}$ decouple in the log likelihood, leading to simple updates in both the E and M steps. While this is not a conditional likelihood

Predicate	Description and evaluation
`stringVal`	Returns string value corresponding to a text span in the statement
`getPhraseMention`	Looks for matching tokens or phrases in a target text, and return true if an exact match is found. e.g., *The subject contains the word postdoc* → `getPhraseMention(subject,stringVal('postdoc'))`
`getPhrasesLike`	Uses an alignment based Textual Entailment (RTE) model to find the closest semantic match for a phrase in a text. Uses distributional semantics to identify semantically similar words. Returns true if a match is found. e.g., *The emails often want me to buy something* → `getPhrasesLike(email,stringVal('buy something'))`
`getSemanticCategory`	Looks for occurrences of pre-specified semantic categories in a target text (identified with Stanford CoreNLP's expanded NER tagger), and returns true if a match is found. e.g., *these emails often have contain prices and quotes* → `getSemanticCategory(body, MONEY)`
`stringMatch`	Returns true if one string value contains another. e.g., *Spam emails are rarely from a yahoo or gmail address* → `not( or(stringMatch(sender, stringVal('yahoo')), stringMatch(sender, stringVal('gmail'))))`
`stringEquals`	Returns true if two string values are equal
`or/ and/ not`	Boolean predicates with usual interpretations
`beginWith/endWith`	Return true if a target text contains a phrase, a similar phrase, or a semantic category at its beginning/end. e.g., *The emails often mention a phone number at the end* → `endWith(body, NUMBER)`
`merge`	Combines multiple elements into a list. e.g., *These emails often refer to problems like baldness and aging.* → `getPhrasesLike(email, merge(stringVal('baldness'), stringVal('aging')))`
`before`	Returns true if there is an instance of one type preceding an instance of another type in a text
`length`	Lengths of lists or text fields (in number of words)
$\geq$, `equals`	Usual arithmetical comparators
`unknown`	Return false by default. Used to deal with statements that cannot be reasonably expressed using predicates in the language. e.g., *These emails are from weird addresses.* → `unknown`

Table 1: Predicates in logical language used by our semantic parser for learning of email based concepts.

(as expected in Section 3.1), in our experiments we found that the NB objective to be empirically effective with our approach.

3.4 Semantic Parsing details

Semantic parsing refers to mapping a sentence s like '*The subject contains the word postdoc*' to a logical form l like `getPhraseMention(subject, stringVal('postdoc'))`. Logical forms can be evaluated in a context x (here, an email) to yield some meaningful output $[\![l]\!]_x$ (whether the statement is true for an email). The predicates (such as `stringVal`) and constants (such as `subject`) come from a pre-specified logical language. Since our focus in this work is concepts about emails, we specify a logical language that is expressive enough to be useful for concept learning in this domain. Table 1 lists the predicates in our logical language along with descriptions of their evaluation, and some illustrative examples showing how they can represent the meaning of natural statements[3]. Note that this logical language can express compositional meanings. e.g., '*These inquiries will usually seek a postdoc opportunity and include a CV*' can be expressed as `and (getPhrasesLike(email, stringVal('seek postdoc opportunity')), (stringMatch attachment (stringVal'CV')))`. The evaluations of some predicates uses NLP tools that go beyond exact keyword matching. In Section 5, we show that it is language understanding (semantic parsing), rather than these resources, which enables learning from natural explanations.

Semantic parsers involve grammars containing mappings from words to symbols in the logical language, as well as coarse syntactic rules. The grammar specifies the possible set of logical interpretations that can be associated with a natural language sentence. For this work, we use CCG based semantic parsing, a popular semantic parsing approach (Zettlemoyer and Collins, 2005; Artzi et al., 2015) that couples syntax with semantics. For the CCG grammar, we manually compile a domain lexicon containing a list of trigger words mapped to their syntactic categories and associated logical predicates. e.g. {'subject', NP, `subject`}. We then use the PAL lexicon induction algorithm (Krishnamurthy, 2016) to expand the lexicon by adding automatically generated entries. For training the parser, we follow the feature set from (Zettlemoyer and Collins, 2007), consisting of indicator features for lexicon entries and rule applications that fire for a given parse of a logical form. We also include

[3]We include a special predicate (`unknown`) to label statements whose meanings go beyond our logical language (last row in Table 1), essentially ignoring them. Such statements compose about 25% of our data. An agent should ideally be able to ask a user about unfamiliar concepts such as 'weird email addresses' that occur in explanations. See Section 6.

string based features denoting the number of words in a string span, and whether a string spans occur at the beginning or end of the utterance. For retrieving the best parses for a statement, we use beam search with a beam size of 500.

While we have chosen a particular instantiation of a semantic parsing formalism, our learning approach is independent of the semantic parsing framework in principle, and only assumes a log-linear parametrization over logical interpretations of sentences. Thus, while we present results for a particular parsing framework and lexicon, the method may conceptually extend to other parsing formalisms such as DCS (Liang et al., 2011).

4 Data

We created a dataset of 1,030 emails paired with 235 natural language statements made by human users in the process of teaching a set of seven concepts. The dataset was collected using the Amazon Mechanical Turk crowdsourcing platform. We deployed two tasks: (i) a *Generation* task requiring workers to create original emails, and (ii) a *Teaching* task requiring workers to write statements that characterize a concept. Below, we describe the data and the two tasks in more detail.

We create an email corpus, rather than use an existing corpus such as Enron, since we wanted diverse examples representative of everyday concepts that most people would be able to understand as well as teach to a computer. Much of the Enron corpus is highly specific and contextualized, making it difficult to teach for an outsider.

The *Generation* task consisted of a web-page resembling a traditional email composition form (with fields: *recipient*, *subject*, *body*, *attachment*), requiring workers to compose emails in a grounded setting. For this task, we recruited 146 workers residing in the United States. The workers were presented with each of the seven concepts in a sequence, where each concept was represented by a short prompt encouraging workers to imagine a scenario (e.g., a boss writing a request to an employee) and write a hypothetical email. See Table 2 for details of email concepts and corresponding prompts. Workers were instructed to be realistic (e.g., to include an attachment if an email is likely to have an attachment in reality), but also creative (to encourage diversity) in composing their emails.

The *Teaching* task was then deployed to collect natural language statements that people would make to teach a particular concept to a machine. Workers were presented with five randomly selected concepts using the same prompts (Table 2) used in the *Generation* task. For each concept, a small sample of emails were shown in a style resembling a traditional email inbox (Figure 4) to illustrate the concept. Half of the emails were from the prompted concept (these emails were highlighted and "starred"), and half were sampled randomly from the other concepts. Workers were encouraged to peruse through the emails while creating up to five statements explaining the concept. A follow-up quiz assessed an understanding of the task, and contributions from workers with low scores were filtered. The final data contains between 30 and 35 statements describing each category.

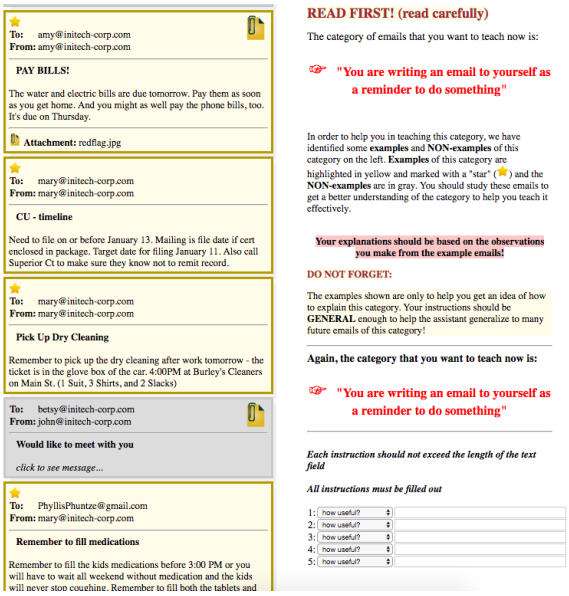

Figure 4: The *Teaching* task used to collect natural language statements characterizing a concept. Each worker is given a concept prompt, together with a set of emails. A turker can enter five statements characterizing the concept.

These emails usually closes with a name or title
Some reminders will have a date and time in the subject
The body of the email may say funny, picture, or internet
Messages to friends sometimes have jpg attachments
Emails from a public domain are not office requests

Table 3: Examples of natural language statements collected from the *Teaching* task

5 Evaluation

In this section, we evaluate the performance of our approach from the perspectives of concept learn-

Concept	# of emails	Prompt
CONTACT	167	"You are writing an email to yourself to personally keep note of a person contact"
EMPLOYEE	149	"You are a boss writing an email to your employee requesting something to be done"
EVENT	138	"You are writing an email to a friend asking to meet up at some event"
HUMOR	134	"You are writing an email to a friend that includes something humorous from the Internet"
MEETING	142	"You are writing an email to a colleague trying to request a meeting about something"
POLICY	146	"You are writing an office email regarding announcement of some new policy"
REMINDER	154	"You are writing an email to yourself as a reminder to do something"

Table 2: Email concepts used in our experiment, together with the prompts used to describe the concept to workers. The same prompt was used in both the *Generation* and *Teaching* tasks.

	CONTACT	EMPLOYEE	EVENT	HUMOR	MEETING	POLICY	REMINDER	Average
BoW	0.510	0.354	0.381	0.484	0.455	0.588	0.415	0.455
BoW tf-Idf	0.431	0.379	0.402	0.513	0.392	0.576	0.399	0.441
Para2Vec	0.238	0.191	0.121	0.252	0.222	0.286	0.092	0.200
Bigrams	0.525	0.385	0.426	0.525	0.458	0.668	0.423	0.487
ESA	0.187	0.209	0.107	0.194	0.154	0.160	0.131	0.164
RTE	0.551	0.353	0.406	0.475	0.398	0.522	0.232	0.419
Keyword filtering	0.521	0.429	0.412	0.425	0.702	0.748	0.392	0.522
LNL-NB	0.628*	0.370	0.453*	0.590*	0.732*	0.878*	0.414	0.581*
LNL-LR	0.608*	0.351	0.568*	0.570*	0.757*	0.898*	0.437	0.598*
LNL-Gold	0.661	0.397	0.677	0.572	0.777	0.917	0.487	0.641
LNI-NB + BoW	0.644	0.409	0.520	0.709	0.723	0.878	0.543	0.632
LNI-LR + BoW	0.634	0.398	0.604	0.704	0.747	0.891	0.567	0.649
LNL-Gold+BoW	0.667	0.449	0.659	0.798	0.771	0.927	0.595	0.695

Table 4: Concept learning performance (F1 scores) using $n = 10$ labeled examples. Columns indicate different concept learning tasks defined over emails. * for the rows corresponding to LNL-NB and LNL-LR denotes statistical significance over the best performing non-LNL model

ing as well as semantic parsing. We first compare our methods against traditional supervised learning methods on the task of learning email-based concepts described in the previous section.

Our baselines include the following models:

Text-only models:

- *BoW:* A logistic regression (LR) classifier over bag-of-words representation of emails
- *BoW tf-idf:* LR classifier over bag-of-words representation, with tf-idf weighting
- *Para2Vec:* LR classifier over a distributed representation of documents, using deep neural network approach by Le and Mikolov (2014).
- *Bigram:* LR model also incorporating bigram features, known to be competitive on several text classification tasks (Wang and Manning, 2012).
- *ESA:* LR model over ESA (Explicit Semantic Analysis) representations of emails (Gabrilovich and Markovitch, 2007), which describe a text in terms of its Wikipedia topics.

Models incorporating Statements:

- *RTE:* This uses a Textual Entailment model (based on features from Sachan et al. (2015)) that computes a score for aligning of each statement to the text of each email. A logistic regression is trained over this representation of the data.
- *Keyword filtering:* Filters based on keywords are common in email systems. We add this as a baseline by manually filtering statements referring to occurrences of specific keywords. Such statements compose nearly 30% of the data. We train a logistic regression over this representation.

Table 4 shows classification performance of our approaches for Learning from Natural Language (*LNL*) against baselines described above for $n = 10$ labeled examples. The reported numbers are average F1 scores over 10 data draws. We observe that *Bigram* and bag-of-word methods are the most competitive among the baselines. On the other hand, *Para2Vec* doesn't perform well, probably due to the relatively small scale of the available training data, while *ESA* fails due to the lack of topical associations in concepts. However, most significantly, we observe that both *LNL-NB* (Naive Bayes) and *LNL-LR* (Logistic Regression) dramatically outperform all baselines for most concepts (except EMPLOYEE), and show a 30% relative improvement in average F1 over other methods ($p < 0.05$, Paired Permutation test). Interestingly, we note that *LNL-NB* and *LNL-LR* show similar performance for most

concepts. For evaluating semantic parsing of natural language statements, we manually annotated the statements in our dataset using the logical language described in Section 3.4. In Table 4, *LNL-Gold* denotes the classification performance with using these annotated gold parses. This corresponds to the hypothetical case where the classifier knows the correct semantic interpretation of each natural language sentence from an oracle. While this provides a further 10% relative improvement over our proposed models, the results suggest that our weakly supervised method is quite effective in interpreting natural language statements for concept learning, without explicit supervision. We also observe that *LNL* models perform significantly better than *Keyword filtering* ($p < 0.05$), indicating that the model leverages the expressiveness of our logical language.

Finally, the last three rows show performance when the *LNL* methods also utilize BoW representations of the data. The further gains over the base *LNL* models suggest that original feature representations and natural language explanations contain complementary information for many concepts.

A significant motivation for this work is the promise of natural language explanations in facilitating concept learning with a relatively small number of examples. Figure 5 shows the dependence of concept learning performance of *LNL(-LR)* on the number of labeled training examples (size of training set). We observe that while our approach consistently outperforms the bag-of-words model (*BoW*), *LNL* also requires fewer examples to reach near optimal performance, before it plateaus. In particular, the generalization performance for *LNL* is more robust than *BoW* for $n < 10$. The performance trajectory for *LNL(-NB)* is similar, and omitted in the figure for clarity.

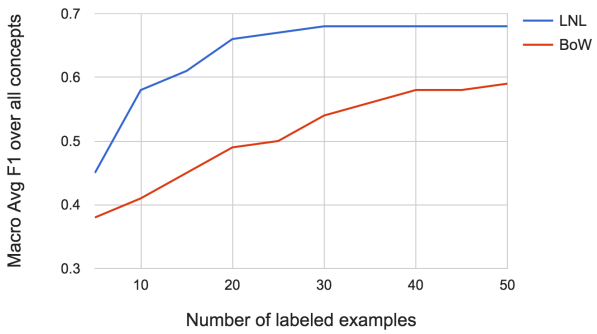

Figure 5: Figure showing Avg F1 accuracy over all concepts vs Number of labeled training examples

	Accuracy
Fully Supervised (ZC07)	0.63
LNL-LR	0.30
LNL-NB	0.28
No training	0.01

Table 5: Semantic parsing performance (exact match) for proposed weakly supervised methods vs full supervision (completely labeled logical forms)

Parsing performance: We next evaluate the parsing performance of our approach, which learns a semantic parser from only concept labels of examples. Table 5 evaluates parsing performance against the gold annotation logical forms for statements. For this task, we check for exact match of logical forms. In the table, full supervision refers to traditional training of a semantic parser using complete annotations of statements with their logical forms (Zettlemoyer and Collins, 2007). The results report average accuracy over 10-fold CV, and demonstrate that while not comparable to supervised parsing, our weakly supervised approach is relatively effective in learning semantic parsers.

Further, exact match to gold annotated logical forms is a restrictive measure. Qualitative analysis revealed that even when the predicted and gold annotation logical forms don't match, predicted logical forms are often strongly correlated in terms of evaluation to gold annotations. e.g., `getPhraseMention( email, stringVal('postdoc'))` vs `getPhraseMention( body, stringVal('postdoc'))`. In about 5% of cases, predicted and gold interpretations are different on the surface, but are semantically equivalent (e.g., `stringEquals( sender, recipient)` vs `stringEquals( recipient, sender)`).

Concept learning vs language interpretation: To delineate the relationship between parsing performance and concept learning more clearly, we plot concept classification performance for different levels of semantic parsing proficiency in Figure 6. For this, we choose the gold annotation logical form for a statement with a probability corresponding to the semantic parsing accuracy, or randomly select a candidate logical form with a uniform probability otherwise for all the statements in our data. The figure shows a (expectedly) strong association between parsing performance and concept learning, although gains from parsing taper after a certain level of proficiency. This is partially explained by the fact that natural statements in our

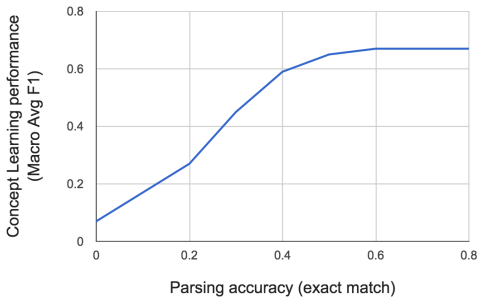

Figure 6: Figure showing concept classification performance vs parsing accuracy

data often contain overlapping information, and that the set of statements in our data set may not be sufficient to achieve perfect classification accuracy.

6 Conclusion and Future Work

We show that natural language explanations can be utilized by supervised learning methods to significantly improve generalization. This suggests a broader class of possible machine learning interfaces that use language to not only expedite learning, but make machine learning accessible to everyday users. Thus, we hope that the current work will inspire further explorations in learning from natural language explanations. In terms of scalability, learning from language would require specification of a logical language and a lexicon of trigger words for each new domain. However, this effort is one-time, and can find re-use across the long tail of concepts in a domain.

A consequence of the expressiveness of language is that in describing a concept, humans often invoke other concepts that may not correspond to existing predicates in the logical language. A natural solution could detect that a feature described in the statement is novel[4], and request the user to teach the unknown concept. The same principle can be applied recursively, resulting in a mixed-initiative dialog, much like between a student and a teacher. Future work can also incorporate other modes of supervision from language. For example, this work ignores modifiers such as 'always' and 'usually', which often carry valuable information that could be incorporated via model expectation constraints.

Acknowledgments

This research was supported by the Yahoo! InMind project. The authors thank Sujay Jauhar and anonymous reviewers for helpful comments and suggestions.

[4] currently the parser returns (unknown) for such statements

References

Yoav Artzi, Kenton Lee, and Luke Zettlemoyer. 2015. Broad-coverage ccg semantic parsing with amr. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1699–1710, Lisbon, Portugal. Association for Computational Linguistics.

Amos Azaria, Jayant Krishnamurthy, and Tom M Mitchell. 2016. Instructable intelligent personal agent. In *AAAI*, pages 2681–2689.

Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *EMNLP*, volume 2, page 6.

Alan W Biermann. 1983. Natural language programming. In *Computer Program Synthesis Methodologies*, pages 335–368. Springer.

SRK Branavan, Harr Chen, Luke S Zettlemoyer, and Regina Barzilay. 2009. Reinforcement learning for mapping instructions to actions. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL: Volume 1-Volume 1*, pages 82–90. Association for Computational Linguistics.

SRK Branavan, David Silver, and Regina Barzilay. 2012. Learning to win by reading manuals in a monte-carlo framework. *Journal of Artificial Intelligence Research*, 43:661–704.

Rich Caruana, Nikos Karampatziakis, and Ainur Yessenalina. 2008. An empirical evaluation of supervised learning in high dimensions. In *Proceedings of the 25th international conference on Machine learning*, pages 96–103. ACM.

James Clarke, Dan Goldwasser, Ming-Wei Chang, and Dan Roth. 2010. Driving semantic parsing from the world's response. In *Proceedings of the fourteenth conference on computational natural language learning*, pages 18–27. Association for Computational Linguistics.

Gerald DeJong and Raymond Mooney. 1986. Explanation-based learning: An alternative view. *Machine learning*, 1(2):145–176.

Jacob Eisenstein, James Clarke, Dan Goldwasser, and Dan Roth. 2009. Reading to learn: Constructing features from semantic abstracts. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 958–967. Association for Computational Linguistics.

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of*

the 20th International Joint Conference on Artifical Intelligence, IJCAI'07, pages 1606–1611, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Kuzman Ganchev, Jennifer Gillenwater, Ben Taskar, et al. 2010. Posterior regularization for structured latent variable models. *Journal of Machine Learning Research*, 11(Jul):2001–2049.

Dan Goldwasser and Dan Roth. 2014. Learning from natural instructions. *Mach. Learn.*, 94(2):205–232.

David Harel, Assaf Marron, and Gera Weiss. 2012. Behavioral programming. *Commun. ACM*, 55(7):90–100.

Jayant Krishnamurthy. 2016. Probabilistic models for learning a semantic parser lexicon. In *Proceedings of NAACL-HLT*, pages 606–616.

Jayant Krishnamurthy and Tom M Mitchell. 2012. Weakly supervised training of semantic parsers. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 754–765. Association for Computational Linguistics.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338.

Quoc V. Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pages 1188–1196.

Percy Liang, Michael I Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 91–99. Association for Computational Linguistics.

Percy Liang, Michael I Jordan, and Dan Klein. 2011. Learning dependency-based compositional semantics. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 590–599. Association for Computational Linguistics.

Percy Liang and Christopher Potts. 2015. Bringing machine learning and compositional semantics together. *Annu. Rev. Linguist.*, 1(1):355–376.

Gideon S Mann and Andrew McCallum. 2010. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of machine learning research*, 11(Feb):955–984.

Tom M Mitchell, Richard M Keller, and Smadar T Kedar-Cabelli. 1986. Explanation-based generalization: A unifying view. *Machine learning*, 1(1):47–80.

Mrinmaya Sachan, Kumar Dubey, Eric P Xing, and Matthew Richardson. 2015. Learning answer-entailing structures for machine comprehension. In *ACL (1)*, pages 239–249.

Sida Wang and Christopher D. Manning. 2012. Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 90–94, Stroudsburg, PA, USA. Association for Computational Linguistics.

Luke S. Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI '05, Proceedings of the 21st Conference in Uncertainty in Artificial Intelligence, Edinburgh, Scotland, July 26-29, 2005*, pages 658–666.

Luke S Zettlemoyer and Michael Collins. 2007. Online learning of relaxed ccg grammars for parsing to logical form. In *EMNLP-CoNLL*, pages 678–687.